FACE DETECTION SYSTEM

Miss. Saie Suraj Vadke[1], Miss. Natasha Anil Jangale[2], Miss. Pallavi Bhaskarao Sabale[3], Prof. S. V. Phulari[4]

*[1,2,3]Student, Department of Computer Engineering,
Pune District Education Association College of Engineering Manjari Hadapsar Pune.*

[4]Guide, Department of Computer Engineering, Pune District Education Association College of Engineering Manjari Hadapsar Pune.

ABSTRACT

Face (facial) recognition is the identifification of humans by the unique characteristics of their Faces.Face recognition technology is the least intrusive and fastest biometric technology. It works with the most obvious individual identififier, the human face. With increasing security needs and with advancement in technology extracting information has become much simpler. This project aims on building an application based on face recognition using different algorithms and comparing the results. The basic purpose being to identify the face and retrieve information stored in the database. It involves two main steps. First to identify the distinguishing factors in image n storing them and Second step to compare it with the existing images and returning the data related to that image.

I. INTRODUCTION

There have been many attempts to solve the human face detection problem. The early approaches are aimed for gray level images only, and image pyramid schemes are necessary to scale with unknown face sizes. View-based detectors are popular in this category, including Rowley's neural networks classifier, Sung and Poggio's

correlation templates matching scheme based on image invariants and Eigen-face decomposition. Model based detection is another category of face detectors. For color images, various literatures have shown that it is possible to separate human skin regions from complex backgrounds based on either YCbCr or HSV color space. The face candidates can be generated from the identified skin regions. Numerous approaches can be applied to classify face and non-face from the face candidates, such as wavelet packet analysis, template matching for faces, eyes and mouths, feature extraction using watersheds and projections. In this project, a new face detector for color images is developed. The objective of this project is to develop a very efficient algorithm, in terms of low computational complexity, with the maximum number of face detections and the minimum number of false alarms. To achieve these objectives first, the image is transformed to HSV color space, where the skin pixels are determined. The skin regions in HSV space are described by crossing regions of several 3D linear equations, which are found using training data. Also, the median luminance condition Y value of the image is determined. For high luminance images, the chances that more non-skin pixels are set to skin regions are high, thus an additional but simple classification on YCbCr space is performed to remove hair pixels. Hence, a binary mask of the original image can be obtained. This binary mask is then filtered with some image morphology processing to break connections between faces and remove scattered noise. A connected component analysis is followed to determine the face candidates. The final step is to determine real faces from the face candidates using a multilayer classification scheme. The application of this project justifies an assumption that the faces will have approximately the same size.

The face detection system can be divided into the following steps:-

1. Pre-Processing: To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples, that is the face images are obtained by cropping images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

2. Classification: Neural networks are implemented to classify the images as faces or non faced by training on these examples. We use both our implementation of the neural network and the Matlab neural network toolbox for this task. Different network configurations are experimented with to optimize the results.

3. Localization: The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has been done on:- Position Scale Orientation Illumination.

II. METHODOLOGY

A design methodology to accelerate the face detection for embedded systems is described, starting from algorithm optimization (high level) and ending with software and hardware codesign (low level) by addressing the issues and the design decisions made at each level based on the performance measurements and system limitations. The implemented embedded face detection system consumes very little power compared with the traditional PC software implementations while maintaining the same detection accuracy. The proposed face detection acceleration methodology is suitable for real time applications.

System Overview: The system consists of a frontend and a backend application. In this chapter, we will explain the architecture behind the platform and the libraries used to implement the face detection and recognition system. Architecture We will be using the MVC framework, therefore the software's source code will be separated into three layers as shown below.

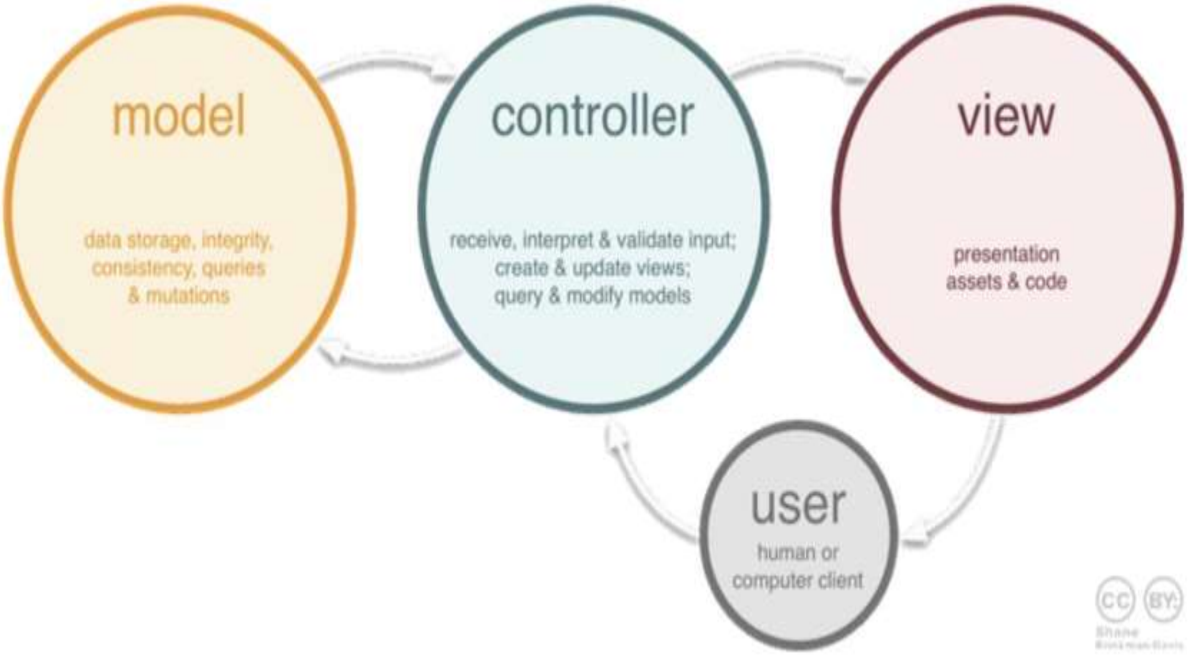
model
data storage, integrity, consistency, queries & mutations
controller
receive, interpret & validate input; create & update views; query & modify models
view
presentation assets & code
user
human or computer client
CC BY
Shane

Fig. Model-View-Controller Framework

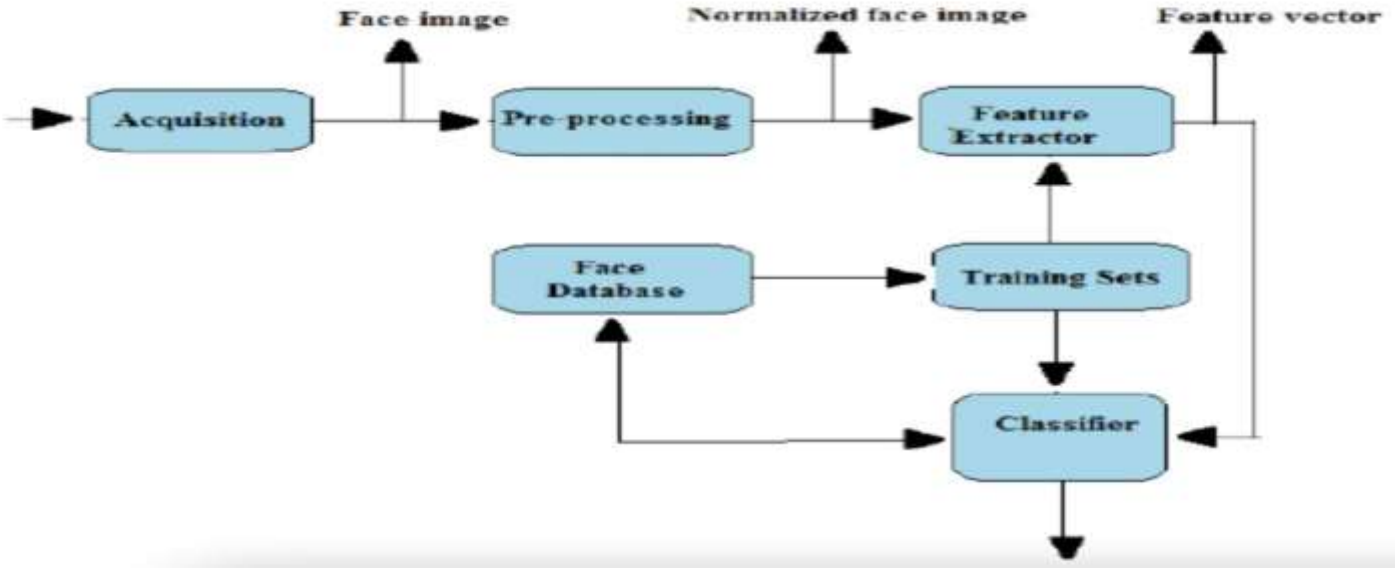

Fig. Block Diagram Of face detection system

FACE DETECTION MODELS

The system we built on Node.js will make use of the following three face detection models.

1. SSD MobileNet v1

One of the main face detectors that is going to be used as a model is a SSD (Single Shot Multibox Detector), which is consists of a CNN based on MobileNet V1; however, it includes additional box prediction layers that are stacked on the top of the network. The neural network can compute the locations of each face in an image and then it will return the bounding boxes and its probability for each face found. This detector is focused on obtaining high accuracy in detecting face bounding boxes instead of low inference time. SSD MobileNet is a face detection model that has been pretrained on the WIDERFACE dataset which is a publicly available.

2. Tiny Face Detector:

The face recognition API implements an optimized face detector called Tiny Face Detector, this detector uses depthwise separable convolutions instead of regular convolutions, which makes it much faster than the other mentioned detectors. However, it is slightly less accurate face detector compared to the SSD MobileNet V1. The Tiny Face Detector has a really good performance of detection, making it much faster, smaller and less resource consuming compared to the SSD MobileNet V1 face detector model. This model is highly optimized for mobile and web interfaces. The model is trained on a customized dataset of ~14K labeled using bounding boxes. Moreover, the face model has been trained to predict bounding boxes with complete cover facial feature points, therefore it produces better results on face landmark detections than SSD MobileNet V1.

3. MTCNN (Multi-task Cascaded Convolutional Neural Networks)

Lastly, there is also another face detector implemented called MTCNN (Multi-task Cascaded Convolutional Neural Network) which is used for experimental purposes. This face model can detect a wide range of face bounding box sizes. However, it consumes a lot of resources and it is not completely optimized. MTCNN is a 3-stage cascaded CNN, which simultaneously returns 5 face landmark points with bounding boxes and scores for each face.The MTCNN will return the bounding boxes for each face and it will also include a probability score from (0.00 to 1.0). The scores can be used to filter the bounding boxes as a tolerance value [3].

III. WORKING

Face detection applications use algorithms and ML to find human faces within larger images, which often incorporate other non-face objects such as landscapes, buildings and other human body parts like feet or hands. Face detection algorithms typically start by searching for human eyes -- one of the easiest features to detect. The algorithm might then attempt to detect eyebrows, the mouth, nose, nostrils and the iris. Once the algorithm concludes that it has found a facial region, it applies additional tests to confirm that it has, in fact, detected a face. To help ensure accuracy, the algorithms need to be trained on large data sets incorporating hundreds of thousands of positive and negative images. The training improves the algorithms' ability to determine whether there are faces in an image and where they are. The methods used in face detection can be knowledge-based, feature-based, template matching or appearance based. Each has advantages and disadvantages:

- Knowledge-based, or rule-based methods, describe a face based on rules. The challenge of this approach is the difficulty of coming up with well-defined rules.
- Feature invariant methods -- which use features such as a person's eyes or nose to detect a face -- can be negatively affected by noise and light.
- Template-matching methods are based on comparing images with standard face patterns or features that have been stored previously and correlating the two to detect a face. Unfortunately these methods do not address variations in pose, scale and shape.
- Appearance-based methods employ statistical analysis and machine learning to find the relevant characteristics of face images. This method, also used in feature extraction for face recognition, is divided into sub-methods.digital cameras.
- Neural networks have been used to recognize patterns in videos and images, such as faces, vehicles, anomalies in medical images, and interesting patterns in other types of collected data.
- Development of multiple face recognition (MFR) systems is useful for implementing applications such as bank security management systems. Face recognition has several applications such as in the field of security, forensic and requires more accuracy and reliability.

V. OBJECTIVES

- Implement face recognition in an optimum way in terms of runtime onto a web application and run it and optimize it for embedded systems.
- Various algorithms and methodologies are studied, and hardware resources planning will be done to achieve the goal.
- Compare different face recognition models based on performance and accuracy. Use neural networks to map the characteristics of a human face to a face descriptor, also sometimes referred as face embeddings.
- Compute the Euclidean distance between two face descriptors and decide whether two faces are similar based on a threshold value.

VI. RESULT AND DISCUSSION

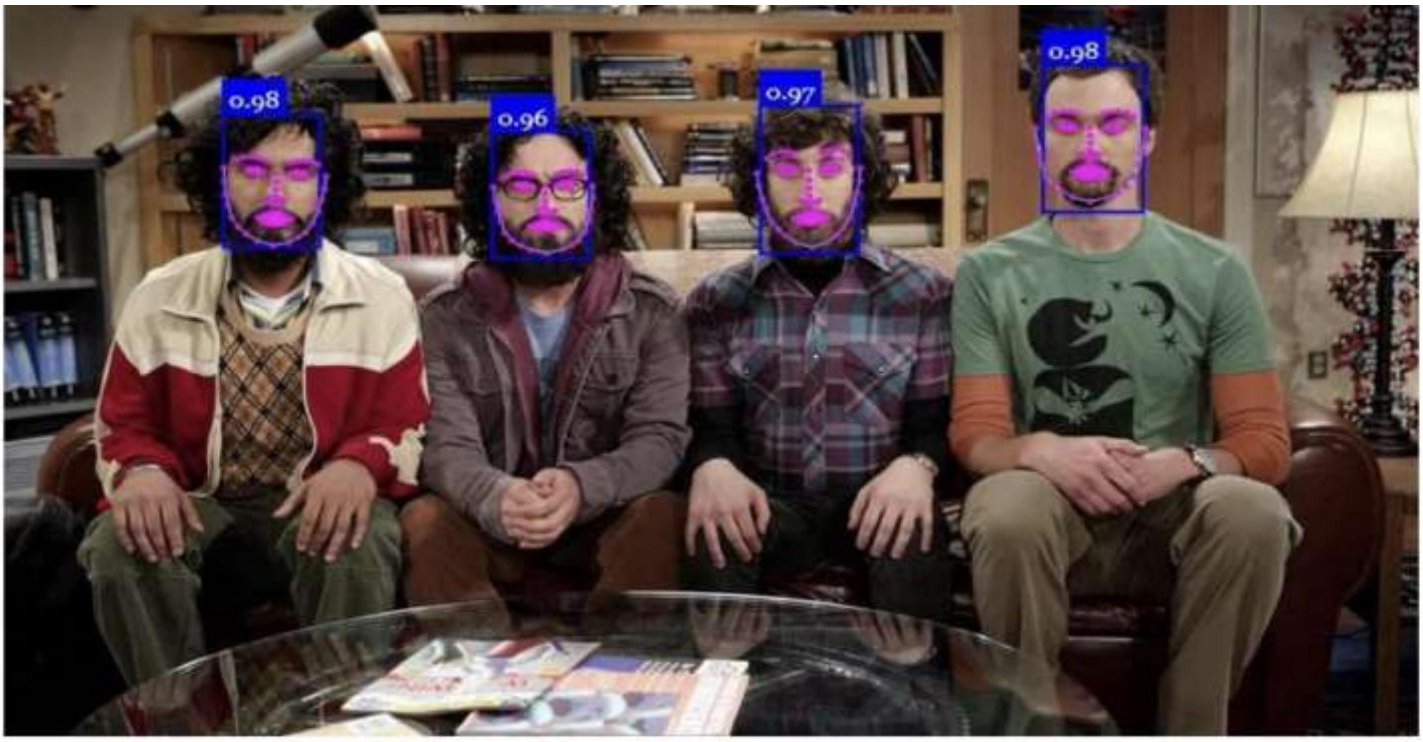

Fig. Face Detection

The results demonstrate that using deep neural networks and the latest techniques can improve the prediction accuracy for face detection, face recognition, and face extraction. In particular, insights that arise from experimenting, understanding, and interacting with neural networks led us to the most valuable observations.

Enabling other students to freely explore this compelling, but sometimes intimidating, field is perhaps the central focus of this work. Consecutively, we need additional tools to allow us to explore deep learning in terms of algorithms, architecture and design techniques.

VII. CONCLUSION

In this paper, we studied the use of deep neural networks on face detection and recognition. We also analyzed some of the different neural networks including convolutional neural networks, depthwise separable convolutions, and densely connected convolutional networks. We have implemented a simple face detection and recognition system using Node.js and TensorFlow.js core API making use of existing API like the face-api.js package located at https://github.com/justadudewhohacks/face-api.js/. This library is optimized to be used inthe browser. Generating the data is out of the scope of this research, but there are publicly available datasets that can be used to load in the app for doing face recognition. The API allowed us to run different experiments where users can detect the emotion of a person, the gender, and the age of a person. In the last years, facial recognition has been well developed and now we can build apps that can recognize faces with great accuracy. The fascinating part about this experiment is to find more useful applications for this technology. As we enter the top era of big data, complex systems require more and more data. The diverse training data that we will use has to be representative of the data that the neural network will come across in the real world. Along with the training data, it is difficult to understand the weights that the neural network has learned. Nonetheless, when using multiple layers, it could be really difficult to draw conclusions from the weights related to the decision that a neutron is actually handling. This paper has covered the core concepts used in deep learning i.e. what mechanisms of backend computation result in enhanced model accuracy for face detection and face recognition. A real thanks to the researchers in this field whose discoveries and findings have contributed to the true power of neural networks.

VIII. REFERENCES

1. *S. Min, B. Lee, and S. Yoon, “Deep learning in bioinformatics,” Briefings in Bioinformatics, 2016.*
2. *G. Dahl, N. Jaitly, “Multi-task Neural Networks for QSAR Predictions,” University of Toronto, 2014.*
3. *K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499- 1503, Oct. 2016. doi: 10.1109/LSP.2016.260334*
4. *S. Linnainmaa, “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors,” Master’s thesis, Univ. Helsinki, 1970.*
5. *P. J. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” PhD thesis, Harvard University, 1974.*

6. *J. Yu and C. Li, "Face Recognition Based on Euclidean Distance and Texture Features," 2013 International Conference on Computational and Information Sciences, Shiyang, 2013, pp. 211- 213. doi: 10.1109/ICCIS.2013.63*
7. *T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724.*